Applying Text Embedding Models for Efficient Analysis in Labeled Property Graphs

Michal Podstawski

NASK National Research Institute
Warsaw, Poland

Abstract. Labeled property graphs often contain rich textual attributes that can enhance analytical tasks when properly leveraged. This work explores the use of pretrained text embedding models to enable efficient semantic analysis in such graphs. By embedding textual node and edge properties, we support downstream tasks including node classification and relation prediction with improved contextual understanding. Our approach integrates language model embeddings into the graph pipeline without altering its structure, demonstrating that textual semantics can significantly enhance the accuracy and interpretability of property graph analysis.

1 Introduction

Labeled property graphs (LPGs) represent a flexible data model in which nodes and edges are not only connected through relationships but also enriched with key-value properties, most of which are textual [1]. This structure is well-suited to domains such as knowledge graphs, social networks, and institutional data, where descriptive attributes carry important semantic information. Despite this, most analytical approaches to LPGs focus primarily on structural information - such as connectivity patterns or edge types - while treating textual fields as flat labels or ignoring them altogether.

Meanwhile, recent developments in natural language processing have produced powerful pretrained text embedding models capable of capturing nuanced semantic meaning from unstructured text. These models, when applied to LPGs, offer a simple yet effective means of enriching analysis with language-level understanding. Text embeddings can serve as dense, general-purpose representations of nodes and relationships, without requiring any modifications to the underlying graph structure.

In this work, we investigate how text embedding models can be used to enhance property graph analysis. Specifically, we apply them to two fundamental tasks: node classification and link prediction. In both cases, textual attributes are transformed into semantic vector representations using a state-of-the-art embedding model. These embeddings are then used as inputs to lightweight classifiers, enabling the graph to be analyzed through its language rather than just its structure.

The results suggest that integrating textual semantics into LPG workflows is a practical and scalable strategy for improving the accuracy and interpretability of graph-based analysis.

2 Related work

Graph-based machine learning methods such as GCNs [2], GraphSAGE [3], and R-GCN [4] have shown strong performance in tasks like node classification and link prediction, but they largely ignore textual node and edge attributes. Meanwhile, pretrained text embedding models [5, 6] capture rich semantic information from text and are widely used in NLP tasks.

Recent work at the intersection of graphs and language includes combining structural and textual features in graphs [7, 8], but these methods often assume fixed schemas or require joint training. Property graphs, with their flexible attributes, remain underexplored in this context.

Our approach differs by integrating pretrained text embedding models into labeled property graph analysis without modifying the graph structure. This enables semantically informed classification and relation prediction using existing text fields as-is.

3 Datasets

We use publicly available labeled property graph datasets provided by Neo4j [9], a widely adopted graph database platform. Used datasets represent diverse domains and graph structures:

- Twitter Trolls (nodes 281136, edges 493160) [10]: A social network graph capturing interactions between accounts linked to coordinated disinformation campaigns.

- Legis (nodes 11825, edges 523004) [11]: A knowledge graph representation of the U.S. Congress comprising legislators, bills, committees, votes, and related entities.

- WWC 2019 (nodes 2486, edges 14799) [12]: A sports-focused graph dataset modeling the 2019 FIFA Women's World Cup, including players, teams, matches, and events.

- Stack Overflow (nodes 6193, edges 11540) [13]: A graph modeling Stack Overflow questions, answers, tags, comments, and the relationships between them.

These datasets offer rich textual properties on nodes and relationships, making them well-suited for evaluating the integration of text embedding models into graph analysis tasks.

4 Solution

Our approach integrates pretrained text embedding models into the analysis of labeled property graphs, enabling semantic-aware workflows for two tasks: node classification and relation prediction. We employ the Qwen3-Embedding-0.6B model [16], a compact but high-performing text embedding model from Alibaba, which achieves state-of-the-art results on the Massive Text Embedding Benchmark (MTEB) [14, 15] despite its small size. This model allows us to efficiently embed textual node and edge properties into dense vector representations without modifying the structure of the graph or fine-tuning the model.

We formulate two prediction tasks based on language-model embeddings of graph nodes. In both cases, the available textual and relational information for node is serialized into a coherent string and processed by the embedding model to produce a 1024-dimensional vector representation. Instances are randomly split into training and test sets, with 90% used for training and 10% reserved for evaluation. Standard classifiers - Random Forest, Logistic Regression, SGDClassifier, and Support Vector Machine (SVM) - are trained on the embeddings and evaluated on the held-out test set.

In the node label prediction task, the input string contains the textual properties of the node, combined into a single coherent description. The classifiers learn to map these embeddings to the corresponding node labels and can subsequently be applied to unlabeled nodes.

In the relation prediction task, the goal is to infer missing factual information (Figure 1). For each labeled instance, a specific relation of the source node is withheld (for example, the relation between a player and a team). The remaining relations of the source node, together with its connected neighbor nodes, are included in the textual description. The trained classifiers predict the correct target node, thereby recovering the withheld relation.

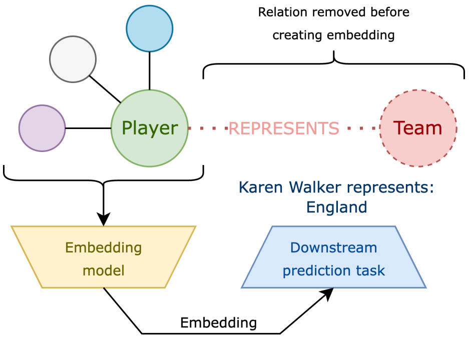
Relation removed before creating embedding
Player
REPRESENTS
Team
Karen Walker represents: England
Embedding model
Downstream prediction task
Embedding

Fig. 1: **Relation Prediction Setting:** In this setting, a specific relation of the source node (e.g., between a *Player* and a *Team* in the WWC 2019 dataset) is withheld prior to embedding generation. The embedding model encodes the source node based on its remaining relations and neighboring nodes. The resulting embedding is then used in a downstream classification task to predict the correct target node, thereby recovering the withheld relation.

All textual inputs are normalized to ensure consistency and kept within the token limits of the model. Embeddings are precomputed and cached to enable efficient training and inference. This method requires no modification of the original property graph and remains model-agnostic, allowing substitution with larger or domain-specific embedding models if needed. Overall, the approach allows us to leverage the semantic richness of textual properties in labeled property graphs, enabling more expressive and accurate analysis through modern language model embeddings.

5 Results

The proposed approach was applied to property graphs containing varied node types and rich textual attributes. In the node classification task, generated embeddings enabled accurate prediction of labels. Results are presented in Table 1. For relation prediction, the method successfully recovered information carried by removed edges, based solely on node descriptions and local context. Representative results are gathered in Table 2.

The results were consistently strong across datasets, with high classification accuracy and clearly meaningful relation predictions. The model handled both sparse and dense text fields effectively and showed robustness to inconsistencies or missing attributes. Overall, the combination of text embeddings and labeled property graph structure proved to be an effective and efficient foundation for semantic graph analysis.

Table 1: Classifier performance for predicting node labels across datasets.

Classifier	Accuracy	Precision	Recall	F1 Score
Random Forest	0.992	0.992	0.992	0.990
Logistic Regression	0.993	0.993	0.993	0.991
SGDClassifier	0.995	0.995	0.995	0.994
Support Vector Machine	1.000	1.000	1.000	1.000

(a) WWC 2019

Classifier	Accuracy	Precision	Recall	F1 Score
Random Forest	0.984	0.984	0.984	0.983
Logistic Regression	0.999	0.999	0.999	0.999
SGDClassifier	0.994	0.993	0.994	0.994
Support Vector Machine	1.000	1.000	1.000	1.000

(b) Twitter Trolls

Classifier	Accuracy	Precision	Recall	F1 Score
Random Forest	0.822	0.823	0.822	0.821
Logistic Regression	0.926	0.926	0.926	0.926
SGDClassifier	0.928	0.930	0.928	0.929
Support Vector Machine	0.939	0.939	0.939	0.939

(c) Stack Overflow

Table 2: Classifier performance for link prediction across datasets.

Classifier	Accuracy	Precision	Recall	F1 Score
Random Forest	0.948	0.954	0.948	0.945
Logistic Regression	0.883	0.856	0.883	0.847
SGDClassifier	0.979	0.982	0.979	0.978
Support Vector Machine	0.998	0.998	0.998	0.998

(a) WCC 2019 - `REPRESENTS` relation

Classifier	Accuracy	Precision	Recall	F1 Score
Random Forest	0.796	0.807	0.796	0.786
Logistic Regression	0.858	0.875	0.858	0.851
SGDClassifier	0.919	0.914	0.919	0.916
Support Vector Machine	0.845	0.862	0.845	0.837

(b) Legis - `IS_MEMBER_OF` relation

Classifier	Accuracy	Precision	Recall	F1 Score
Random Forest	0.987	0.987	0.987	0.987
Logistic Regression	0.993	0.993	0.993	0.993
SGDClassifier	0.987	0.987	0.987	0.987
Support Vector Machine	0.987	0.987	0.987	0.987

(c) Legis - `ELECTED_TO` relation

6 Next Steps

Future work could focus on enhancing the integration of textual and structural information, for example by combining language-model embeddings with topology-aware graph encoders. Another promising direction is task-specific adaptation of the embedding space to capture domain-specific semantics and relation patterns. Scaling to large and dynamic graphs may require efficient incremental embedding updates and distributed processing pipelines. Finally, systematic evaluation across heterogeneous datasets would help to assess generalization and identify domain-dependent optimization strategies.

7 Conclusions

We presented an approach for enhancing labeled property graph analysis using pretrained text embedding models. By encoding textual properties into semantic vectors, we enable more informed classification and relation prediction without modifying the underlying graph structure.

This method is model-agnostic, lightweight, and compatible with existing property graph platforms. Our evaluation demonstrates its broad applicability across domains.

The results highlight the potential of treating textual node and edge attributes as semantically rich signals-supporting more expressive and effective analysis workflows within graph-based systems.

Acknowledgment

This manuscript acknowledges the use of ChatGPT [17], powered by the GPT-4o language model developed by OpenAI, to improve language clarity, refine sentence structure, and enhance overall writing precision.

References

[1] M. Besta, R. Gerstenberger, E. Peter, M. Fischer, M. Podstawski, C. Barthels, G. Alonso, and T. Hoefler, *Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries*, ACM Comput. Surv., vol. 56, no. 2, pp. 1–36, Feb. 2024, doi: 10.1145/3604932.

[2] T. Kipf and M. Welling, *Semi-Supervised Classification with Graph Convolutional Networks*, in ICLR, 2017.

[3] W. Hamilton, Z. Ying, and J. Leskovec, *Inductive Representation Learning on Large Graphs*, in NeurIPS, 2017.

[4] M. Schlichtkrull, T. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, *Modeling Relational Data with Graph Convolutional Networks*, in ESWC, 2018.

[5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, in NAACL, 2019.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient Estimation of Word Representations in Vector Space*, arXiv preprint arXiv:1301.3781, 2013.

[7] L. Yao, C. Mao, and Y. Luo, *KG-BERT: BERT for Knowledge Graph Completion*, arXiv preprint arXiv:1909.03193, 2019.

[8] X. Wang, T. Lv, L. Lin, Z. Liu, M. Wang, and J. Tang, *KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation*, in TACL, 2021.

[9] Neo4j, Inc., *Neo4j Graph Database*, Neo4j, Inc., https://neo4j.com, 2024.

[10] Neo4j, Inc., *Twitter Trolls Graph Example*, GitHub repository, https://github.com/neo4j-graph-examples/twitter-trolls, 2019.

[11] Neo4j, Inc., *Legis Graph Graph Example*, GitHub repository, https://github.com/neo4j-graph-examples/legis-graph, 2019.

[12] Neo4j, Inc., *Women's World Cup 2019 Graph Example*, GitHub repository, https://github.com/neo4j-graph-examples/wwc2019, 2019.

[13] Neo4j, Inc., *Stack Overflow Questions, Answers, Tags, and Comments Graph Example*, GitHub repository, https://github.com/neo4j-graph-examples/stackoverflow, 2019.

[14] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, *MTEB: Massive Text Embedding Benchmark*, in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Dubrovnik, Croatia, May 2023, pp. 2014-2037. Association for Computational Linguistics. DOI: 10.18653/v1/2023.eacl-main.148.

[15] mteb, *MTEB Leaderboard*, Hugging Face Space, accessed July 2025 (originally launched April 2024), https://huggingface.co/spaces/mteb/leaderboard.

[16] Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin, F. Huang, and J. Zhou, *Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models*, arXiv preprint arXiv:2506.05176, 2025.

[17] OpenAI, *ChatGPT*, https://chatgpt.com/, Accessed: 2025-06-14.